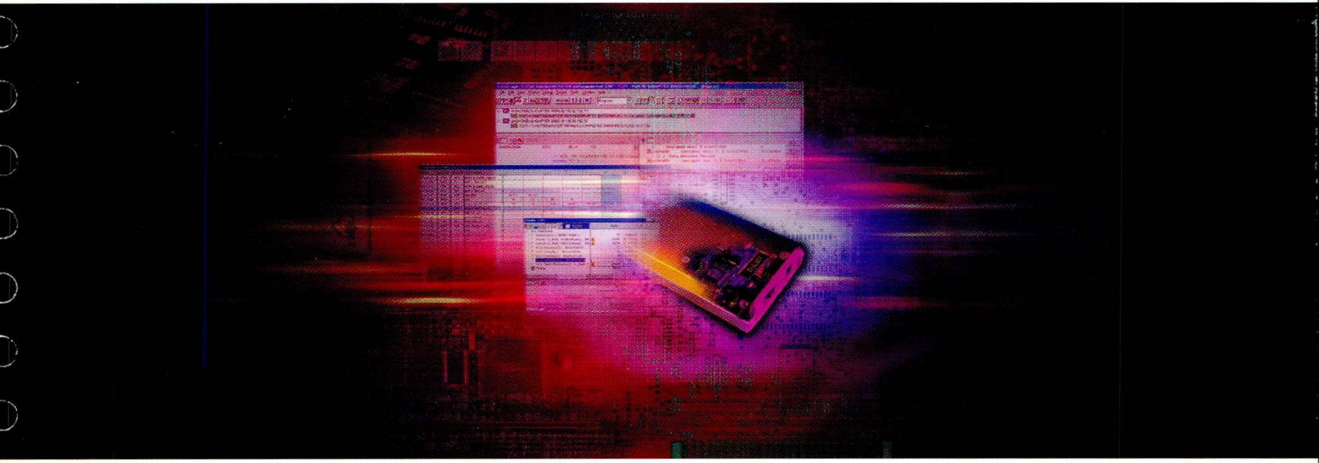
codescape®



Getting Started for DASH3 and DASH4

Getting Started for

DASH3 and DASH4

Cross Products Limited

Legal Notice

IMPORTANT

The information contained in this publication is subject to change without notice. This publication is supplied "as is" without warranty of any kind, either express or implied, including but not limited to the implied warranties or conditions of merchantability or fitness for a particular purpose. In no event shall Cross Products be liable for errors contained herein or for incidental or consequential damages, including lost profits, in connection with the performance or use of this material whether based on warranty, contract, or other legal theory.

This publication contains proprietary information which is protected by copyright. No part of this publication may be reproduced in any form, or stored in a database or retrieval system, or transmitted or distributed in any form by any means, electronic, mechanical photocopying, recording, or otherwise, without the prior permission of Cross Products Limited.

This equipment is Class A Information Technology Equipment (ITE) as defined in EN55022:1994.

Warning

This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

NOTE: THIS EQUIPMENT HAS BEEN TESTED AND FOUND TO COMPLY WITH THE LIMITS FOR "CLASS A" DIGITAL DEVICES, PURSUANT TO PART 15 OF THE FCC RULES. THESE LIMITS ARE DESIGNED TO PROVIDE REASONABLE PROTECTION AGAINST HARMFUL INTERFERENCE WHEN THE EQUIPMENT IS OPERATED IN A COMMERCIAL ENVIRONMENT. THIS EQUIPMENT GENERATES, USES, AND CAN RADIATE RADIO FREQUENCY ENERGY AND, IF NOT INSTALLED AND USED IN ACCORDANCE WITH THE INSTRUCTION MANUAL, MAY CAUSE HARMFUL INTERFERENCE TO RADIO COMMUNICATIONS. OPERATION OF THIS EQUIPMENT IN A RESIDENTIAL AREA IS LIKELY TO CAUSE HARMFUL INTERFERENCE IN WHICH CASE THE USER WILL BE REQUIRED TO CORRECT THE INTERFERENCE AT HIS OWN EXPENSE.

CHANGES OR MODIFICATIONS NOT EXPRESSLY APPROVED BY CROSS PRODUCTS LTD. COULD VOID THE USER'S AUTHORITY TO OPERATE THE EQUIPMENT.

Getting Started for DASH3 and DASH4

Revision History

First release April 2000. CodeScape version 3.0.0

© 2000 Cross Products Limited. All rights reserved.

CodeScape is a registered trademark of Cross Products Limited in the United Kingdom and other countries. All other trademarks or registered trademarks are the property of their respective owners.

Cross Products Ltd
23 The Calls, Leeds, West Yorkshire, LS2 7EH
telephone: +44 113 242 9814
facsimile: +44 113 242 6163
www.crossprod.co.uk
email sales: enquiry@crossprod.co.uk
email support: support@crossprod.co.uk

Contents

How to use this guide 1

Getting started procedure 3

Connection details 5

Configuring the DASH from CodeScape 7

Loading the Extended debug stub 11

Troubleshooting 17

How to use this guide

The *"Getting started procedure"* lists, in chronological order, the tasks you must complete to establish communications between CodeScape, the DASH, and the target using the basic ASE debug stub.

The chapters *"Connection details"* and *"Configuring the DASH from CodeScape"* give you the detailed information required to complete the Getting Started Procedure.

The chapter *"Loading the Extended debug stub"* gives details and examples of the different methods you can employ to load the fully functional Extended debug stub depending on your debugging environment.

System components

The CodeScape 3.0.0 Hitachi SuperH™ development system comprises:

- *Hardware* - DASH hardware and a dongle.
- *Software* - CD containing the CodeScape® development application and online documentation.
- *Documentation* - Getting Started for DASH3 and DASH4, CodeScape user guides, Debug Interface reference manual and DASH Connector Configurations.

Requirements

- *Minimum* - an IBM™ PC or compatible with Pentium® 90MHz processor and 32MB of RAM.
- *Recommended* - an IBM™ PC or compatible with Pentium® II 233MHz processor or above and 128MB of RAM.
- Windows® 95, Windows® 98, Windows NT™ 3.51/4.0, or Windows® 2000.

Getting started procedure

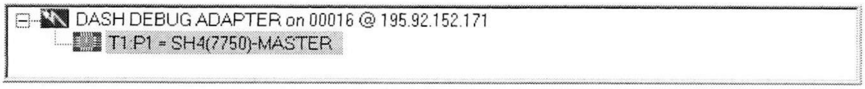
Establishing communications between the DASH and your computer

1. Connect the DASH to a power supply.
2. Connect the DASH to your computer (peer to peer).

   OR

3. Connect the DASH to your network.
4. Check that the “ethernet link OK” and “ethernet data” LEDs on the DASH show activity.
5. Start CodeScape.
6. Configure the DASH’s IP address using CodeScape.
7. Confirm that CodeScape reports the DASH is present in the Target region.

Figure 1: Target region in CodeScape showing that a DASH is present (SH4 system).



```
DASH DEBUG ADAPTER on 00016 @ 195.92.152.171
    T1:P1 = SH4(7750)-MASTER
```

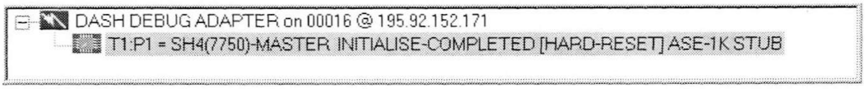
Establishing communications between the DASH and the target

1. Exit CodeScape and power off the DASH.
2. Connect the DASH to the target.
3. Power up the target.
4. Power up the DASH.

5. Start CodeScape.

   Confirm that CodeScape reports the DASH is present in the Target region and shows the status of the target processor connected to the DASH.

   Figure 2: Target region in CodeScape showing a DASH and the status of the target processor.

   ```
   DASH DEBUG ADAPTER on 00016 @ 195.92.152.171
       T1:P1 = SH4(7750)-MASTER INITIALISE-COMPLETED [HARD-RESET] ASE-1K STUB
   ```

   This means that communications have been established between your computer, the DASH, and the target and that the ASE debug stub has been loaded onto the target. You can now configure the DA Start-up options to load the Extended debug stub.

6. If CodeScape reports that the DASH is not communicating with the target, you might need to configure the JTAG clock frequency and restart CodeScape.

Connection details

WARNING: Never connect a DASH4 to an SH3 target or vice versa, this can result in permanent damage to the DASH and the target processor.

Power supply

The DASH requires a 9-16 volts DC power supply of minimum 100 mA rating. The DC input to the DASH is polarity protected.

Connecting to the target

The DASH connects to the target via a 14 way IDC connector. A shielded IDC to IDC cable is supplied. To fully comply with EC directive 89/336/EEC concerning emissions (EN55022) and immunity (EN50082-1) you must connect the flying lead on the supplied cable to a suitable secure earthing point on your target system.

You can make your own cable if required. Pin details are given for all supported Hitachi SuperH™ processors in the "*DASH Connector Configurations*" document. This is a JTAG interface with a theoretical maximum length of 30cm based on a 20MHz clock speed. We recommend that the cable does not exceed 25cm in length and the on-board track length to the processor does not exceed 5cm. You may be able to achieve longer cable lengths at slower clock speeds.

Connecting to the network

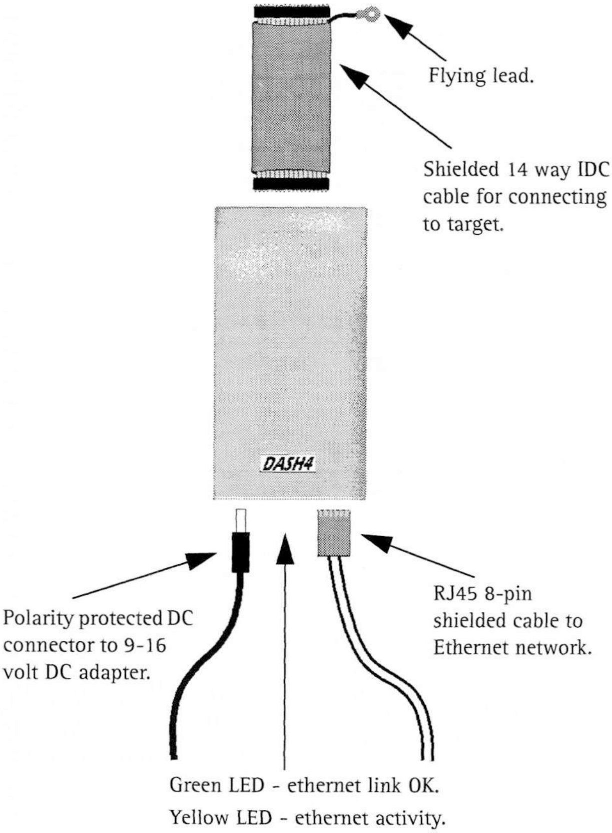
The DASH connects to a network via an RJ45 8-pin shielded ethernet cable with a maximum length of 3m.

You can connect the DASH directly to your computer (peer to peer) by connecting an RJ45 cable directly to your computer's network card. A peer to peer connection requires a crossed RJ45 cable, whereas a connection to a network requires a straight through cable.

The RJ45 connector at the DASH end must have the following pin configuration:

RJ45 Pin	Function
1	TXD+
2	TXD-
3	RXD+
6	RXD-

Figure 1: Cable connections on the DASH


Flying lead.
Shielded 14 way IDC cable for connecting to target.
DASH4
Polarity protected DC connector to 9-16 volt DC adapter.
RJ45 8-pin shielded cable to Ethernet network.
Green LED - ethernet link OK.
Yellow LED - ethernet activity.

Configuring the DASH from CodeScape

Installing CodeScape

1. Insert the CodeScape CD into your computer's CD-ROM drive. Setup should start automatically. If Setup does not start automatically, run SETUP.EXE from the CD.
2. Follow the instructions on screen.
3. When CodeScape has installed, go to *CodeScape for DASH* on the *Start, Programs* menu and select *Dongle Driver (Un)Install.*
4. Follow the instructions on screen to install the dongle driver.

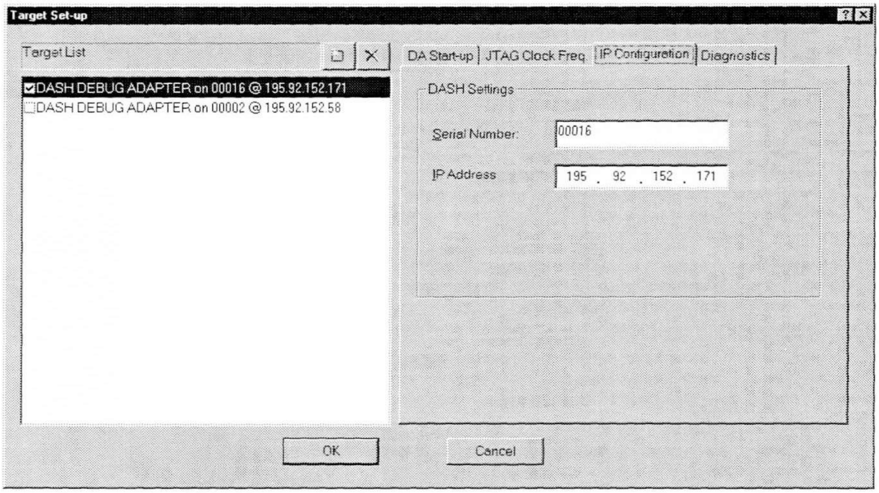
NOTE: *The dongle must always be inserted into the parallel port on your computer when CodeScape is running.*

Configuring the DASH's IP settings

1. Run CodeScape.
2. Click Tools, Configure Target/Communication...
   The Target Set-up dialog box appears.
3. Click ☐.
   The IP Configuration tab appears with a *New DASH Target* to configure.
4. Enter the last five digits of the DASH's serial number. The serial number is given on the base of the DASH.
5. Enter a valid, spare IP Address on your network to assign to the DASH.
   Repeat steps 3 to 5 for each additional DASH(s) that you want to configure.

6. In the Target List, click the check box next to the DASH(s) that you want to connect to.
7. Click OK.
   CodeScape restarts and scans the network for all the DASHs in the list with a check mark.

Figure 2: Target Set-up dialog box showing two DASHs.



☑ DASH DEBUG ADAPTER on 00016 @ 195.92.152.171
☐ DASH DEBUG ADAPTER on 00002 @ 195.92.152.58

Serial Number: 00016

IP Address: 195 . 92 . 152 . 171

Notes:

1. *If CodeScape finds the specified DASH at a different IP Address, CodeScape asks if you want to change the IP address.*
2. *If the DASH is in use by CodeScape on another computer, CodeScape displays a message that the device is not available. If this happens shut down CodeScape on the unwanted computer and power cycle the DASH.*
3. *If you encounter communications problems with the DASH or your network, you can get detailed diagnostics information on the Diagnostics tab on the Target Set-up dialog box.*

Setting the JTAG clock frequency

The JTAG clock frequency (TCK) is preset to 20MHz which is suitable for most SH4 and SH3 applications.

If you have specific requirements you can set the JTAG clock frequency using the Target Set-up dialog box so that it matches the requirements of the target hardware you are debugging.

The following general rules apply:

- SH4 - the JTAG clock frequency must be less than the target hardware's peripheral clock frequency.
- SH3 - the JTAG clock frequency must be less than half the target hardware's CPU clock frequency.

NOTE: *Refer to Hitachi documentation to determine the correct JTAG clock frequency to use for your target hardware.*

Loading the Extended debug stub

The ASE debug stub is a minimal debug stub loaded into ASE RAM after a target system reset. The purpose of the ASE debug stub is to facilitate basic debugging, system initialization and loading of the larger and functionally complete Extended debug stub.

The Extended debug stub must be loaded into external RAM on the target to make full use of CodeScape's debugging capabilities.

The "*Debug Interface*" reference manual gives detailed information about the ASE and Extended debug stubs and gives example Boot ROM code for specific processors.

Using the DA Start-up options on the Target Set-up dialog box you can specify how to initialize the target's RAM and how to load the Extended debug stub. For these options to take effect you must either power cycle the DASH or do a Hard Reset in CodeScape.

Initializing the target's RAM

There are two methods of initializing the target's RAM:

- Using a Boot ROM.
- Using a Java script.

Loading the Extended debug stub

There are two methods of loading the Extended debug stub:

- As a result of a BIOS call request from the Boot ROM to an address specified in the Boot ROM.
- Forced to load at an address specified on the DA Start-up tab.

Specifying the DA Start-up options

The flowchart on page 14 shows how the options on the DA Start-up tab effect the way the DASH boots up the target. The flow chart should be read in conjunction with these instructions.

On the DA Start-up tab:

1. Select Debug Support Enabled.
   You must select this to get debug data.

NOTE: *The DA Start-up configuration is stored in EEPROM. When debug support is disabled (Debug Support Enabled tick box not checked) and the DASH is power cycled, the DASH would normally not see the target. For this reason, when powering up, the DASH always resets debug support to "enabled". If you want to run CodeScape without debug support, uncheck Debug Support Enabled and do a target Hard Reset from CodeScape.*

Initializing RAM with a Boot ROM

The Boot ROM is run after the ASE debug stub has loaded and is run from address 0xA0000000, the default reset address.

1. Select *Run Boot ROM.*
2. Select *Stub Load Address Supplied by ROM* if your boot ROM includes a BIOS call to load the Extended debug stub. The DA pauses for 25 seconds to allow the BIOS call to be issued by the Boot ROM, otherwise only the ASE debug stub is available.

   OR

   Select *Force Load Extended Stub* and specify where in the target's RAM to load the Extended debug stub.
   The address must be on a long word boundary, that is, the two least significant bits must be zero. If you do not set a load address only the ASE debug stub is available.
3. Select *Halt After Target Reset* to stop your application code after start up.
   If this is left unchecked, program control is returned to the Boot ROM after start up.

Initializing RAM with a script

An example script called examboot.js is provided with CodeScape.

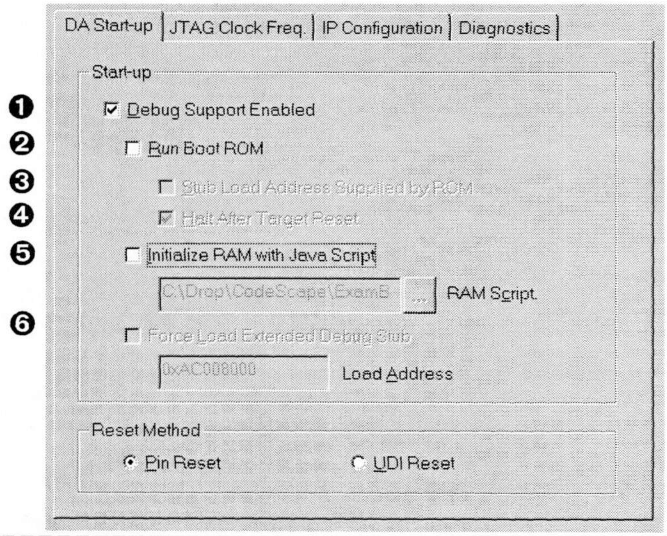
1. Select *Initialize RAM with Java Script.*
   Browse for the script you want to run.
2. Select *Force Load Extended Stub* and specify where in the target's RAM to load the Extended debug stub.
   The address must be on a long word boundary, that is, the two least significant bits must be zero. If you do not set a load address only the ASE debug stub is available.

Reset Method

Select the reset method to specify how the DASH generates a target reset. *Pin Reset* is preferable to *UDI Reset* because it enables the entire target hardware to be reset. Refer to the "*DASH Connector Configurations*" document for pin details.

- Pin Reset uses the DBG_RST# line.
  For this to operate the target system must have a functional DBG_RST# and RESETIN# line available to the DASH. The DASH uses the RESETIN# line as confirmation that the DBG_RST# line assertion occurred correctly.
- UDI Reset can be used if there is no hard-wired reset mechanism on the target hardware. With this selected, upon a hard reset, the DASH sends a serial command on the UDI TDO line to reset the target via JTAG.

Figure 3: DA Start-up tab, numbers show corresponding options on the flowchart



Start-up

☑ Debug Support Enabled

☐ Run Boot ROM

☐ Stub Load Address Supplied by ROM

☑ Halt After Target Reset

☐ Initialize RAM with Java Script

RAM Script: C:\Drop\CodeScape\ExamB

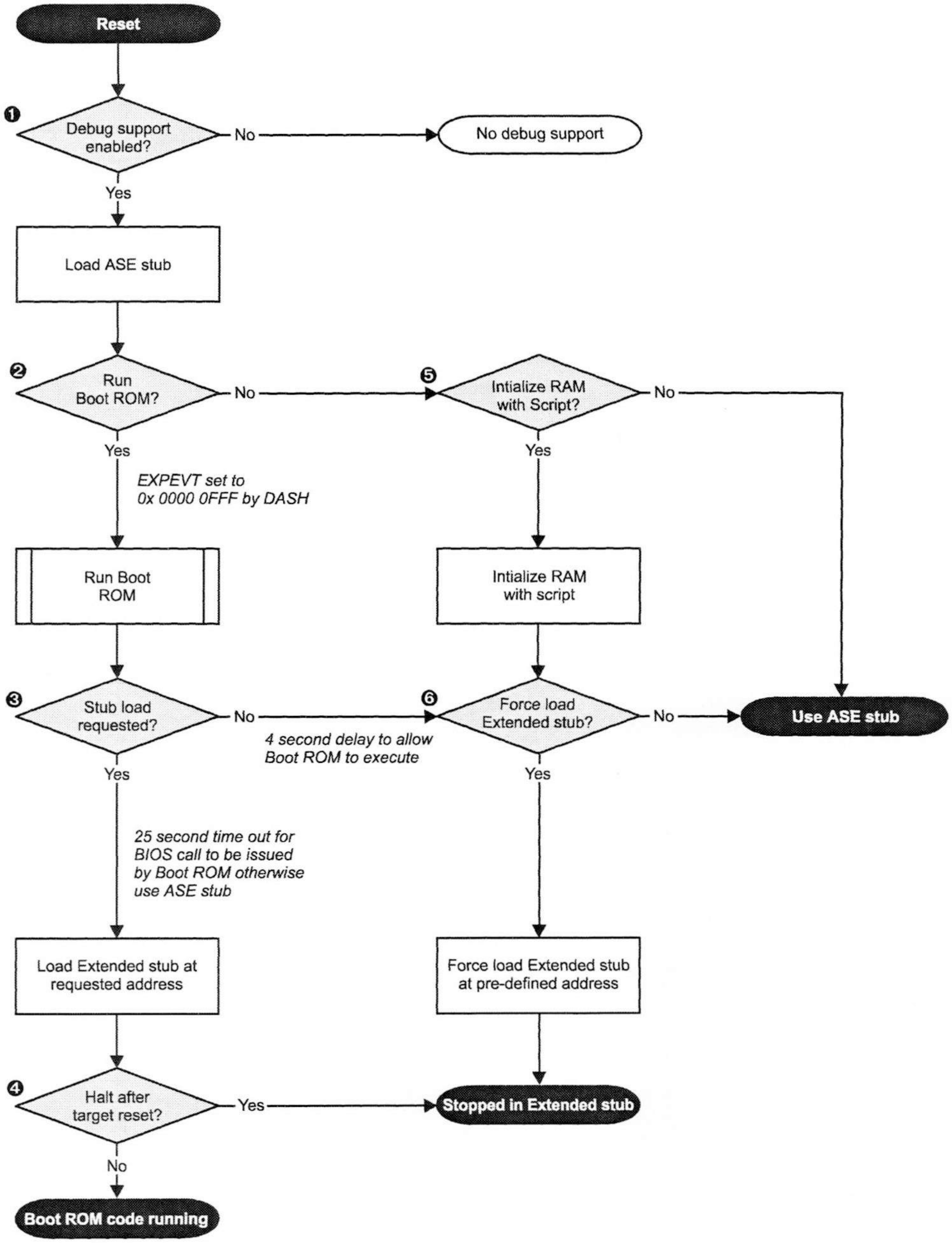
☐ Force Load Extended Debug Stub

Load Address: 0xAC008000

Reset Method

● Pin Reset ○ UDI Reset

Start-up options flow chart


Reset
❶
Debug support enabled?
No
No debug support
Yes
Load ASE stub
❷
Run Boot ROM?
No
❺
Intialize RAM with Script?
No
Yes
Yes
EXPEVT set to 0x 0000 0FFF by DASH
Run Boot ROM
Intialize RAM with script
❸
Stub load requested?
No
4 second delay to allow Boot ROM to execute
❻
Force load Extended stub?
No
Use ASE stub
Yes
Yes
25 second time out for BIOS call to be issued by Boot ROM otherwise use ASE stub
Load Extended stub at requested address
Force load Extended stub at pre-defined address
❹
Halt after target reset?
Yes
Stopped in Extended stub
No
Boot ROM code running

Examples for debugging a new target board

These examples demonstrate how to set up the DA Start-up options depending on your debugging environment. All examples assume the target is fitted with RAM.

Example 1 - Test the new hardware with no Boot ROM fitted.

This configuration allows basic debugging using the only ASE debug stub. In this mode you can verify that the target processor is running. External hardware can be read from, and written to, so that addressing of peripherals can be checked.

DA Start-up dialog configuration:

Debug Support Enabled	Checked
Run Boot ROM	Unchecked
Stub Load Address Supplied by ROM	Unchecked
Halt After Target Reset	Not applicable
Initialise RAM with Java Script	Unchecked
Force Load Extended Stub	Unchecked

Example 2 - Test the new hardware with no Boot ROM fitted. RAM initialized with a script.

This configuration allows basic debugging using only the ASE debug stub. With the addition of a script file the RAM and any peripherals can be initialized. At this point test programs can be loaded into RAM and executed.

DA Start-up dialog configuration:

Debug Support Enabled	Checked
Run Boot ROM	Unchecked
Stub Load Address Supplied by ROM	Unchecked
Halt After Target Reset	Not applicable
Initialise RAM with Java Script	Checked
Force Load Extended Stub	Unchecked

Example 3 - Test the new hardware with a Boot ROM fitted. RAM initialised using a Boot ROM. Boot ROM contains BIOS call.

This configuration provides full debug support using the Extended debug stub. The Extended debug stub is loaded as a result of a BIOS call request from the Boot ROM.

DA Start-up dialog configuration:

Debug Support Enabled	Checked
Run Boot ROM	Checked
Stub Load Address Supplied by ROM	Checked
Halt After Target Reset	Checked
Initialise RAM with Java Script	Unchecked
Force Load Extended Stub	Unchecked

Example 4- Test the new hardware with a Boot ROM fitted. RAM initialised using a Boot ROM. Boot ROM does not contain BIOS call.

This configuration provides full debug support using the Extended debug stub. However, the Extended debug stub is forced to load at a specified address.

DA Start-up dialog configuration:

Debug Support Enabled	Checked
Run Boot ROM	Checked
Stub Load Address Supplied by ROM	Unchecked
Halt After Target Reset	Checked
Initialise RAM with Java Script	Unchecked
Force Load Extended Stub	Checked

Troubleshooting

DASH or target not found

If CodeScape cannot find the DASH or target check that the:

- DASH serial number in the Configure Network Settings dialog box is correct.
- Network cabling between your computer, the DASH and the target is OK.
- Green "ethernet link OK" LED is illuminated on the DASH.
- Yellow "ethernet activity" LED flashes occasionally.
- DASH is not in use by CodeScape on another computer on the network.

NOTE: *If you are having trouble connecting to the DASH over a network, use PING to trace your communications problems or use the diagnostics tab on the Target Set-up dialog box.*

Extended debug stub does not load

When *Stub Load Address Supplied by ROM* is selected on the DA Start-up tab, the DASH expects the Boot ROM to issue a BIOS call to load the Extended debug stub. If the BIOS call is not received within 25 seconds, the reset times out and only the ASE stub is available.

Cannot access target's memory

If you cannot access memory, or areas of memory, on the target you see the message *"No memory range for processor in config files"* when CodeScape starts up, or you see asterisks where there should be data in the Memory window in CodeScape.

Memory ranges and processor timing information are specified in the configuration file, dali.cfg. If you cannot see or access specific areas of memory that you require, you can edit dali.cfg to configure CodesScape with the areas of memory you want to read. If you edit this file, first make a copy of the original file.

Refer to “Creating Project and Configuring the Interface” in the CodeScape user documentation for details of the configuration file.